

Website Vulnerability

Open Web Application Security Project (OWASP) Foundation telah merilis daftar kerentanan umum pada aplikasi berbasis web Top 10 adalah daftar yang dibuat oleh Open Web Application Security Project (OWASP) yang berisi sepuluh jenis kerentanan keamanan aplikasi web paling kritis dan umum terjadi. Daftar ini dibuat berdasarkan penelitian dan analisis mendalam terhadap data yang dikumpulkan dari berbagai sumber di seluruh dunia. Dari daftar ini, kita bisa mengetahui ancaman yang paling memiliki dampak besar atau serius pada keamanan aplikasi web.

Referensi:

<https://www.owasp.org>.

- [A01:2021-Broken Access Control](#)
- [A02:2021 - Cryptographic Failures](#)
- [A03:2021 - Injection](#)
- [A04:2021 - Insecure Design Vulnerabilities](#)
- [A05:2021 - Security Misconfiguration](#)
- [A06:2021 - Vulnerable and Outdated Components](#)
- [A07:2021 - Identification and Authentication Failures](#)
- [A08:2021 - Software and Data Integrity Failures](#)
- [A09:2021 - Security Logging and Monitoring](#)
- [A10:2021 - Server-Side Request Forgery](#)

A01:2021-Broken Access Control

Deskripsi

Broken Access Control adalah salah satu ancaman keamanan aplikasi/web yang paling serius dan menduduki peringkat pertama dalam OWASP Top 10. Kontrol akses adalah proses yang menentukan siapa yang memiliki hak untuk melihat atau menggunakan sumber daya dalam sistem. Kerentanan ini umumnya terjadi ketika sistem gagal menerapkan mekanisme kontrol akses yang memadai diantaranya:

- Tidak terdapat pengecekan akses kontrol dengan memodifikasi URL, *internal application state*, atau *HTML page*, atau menggunakan *custom API attack tool*.
- Mengizinkan *primary key* untuk dapat diganti ke record user lain, yang memungkinkan untuk melihat atau melakukan perubahan pada akun lain.
- Peningkatan sebuah privilege (*Elevation Privilege*), pengguna memiliki akses sebagai admin menggunakan akun *user standard*.
- Manipulasi metadata
- Konfigurasi yang salah pada CORS sehingga menyebabkan API mengakses sistem yang tidak diizinkan

Dampak

- Pengguna yang tidak berwenang dapat mengakses atau mengubah data yang seharusnya dilindungi.
- Pengambilalihan sistem
- Kerugian finansial dan kerusakan reputasi
- Konsekuensi hukum dan regulasi

Mitigasi

- Menolak semua akses kecuali ke direktori yang dapat diakses publik
- Melakukan implementasi mekanisme one time password (OTP) pada seluruh aplikasi sehingga meminimalisir penggunaan CORS.
- Menerapkan kontrol akses minimal pada pengguna (*least user privilege*)
- Menonaktifkan *directory listing web server* dan memastikan file metadata (contohnya .git) dan *file backup* tidak ada di dalam web roots.
- Mencatat kegagalan akses kontrol dan alert admin jika diperlukan (seperti adanya kegagalan yang terjadi berulang - ulang).

- Membatasi ukuran API dan akses ke kontroler untuk meminimalisir kerusakan dari *automated attack tooling*.
- *JWT tokens* harus langsung di hilangkan validasinya pada server setelah logout.
- Melakukan pengujian kerentanan keamanan

A02:2021 - Cryptographic Failures

Deskripsi

Cryptographic Failures, atau kegagalan kriptografi terjadi ketika mekanisme kriptografi yang digunakan untuk melindungi data tidak berfungsi dengan baik. Kegagalan kriptografi disebabkan antara lain:

- Tidak menerapkan enkripsi / pemilihan algoritma yang tidak tepat.
- Kesalahan dalam implementasi
- Manajemen kunci kriptografi yang kurang memadai.
- Pengguna tidak memverifikasi sertifikat elektronik dari server

Dampak

Kegagalan kriptografi dapat menimbulkan risiko keamanan siber yang serius, seperti kebocoran data sensitif, manipulasi data, kerugian finansial, dan kerusakan reputasi bisnis

Mitigasi

- Mengklasifikasikan data yang diproses, disimpan, atau dikirim oleh aplikasi sesuai dengan ketentuan perundang-undangan, persyaratan peraturan, atau kebutuhan bisnis.
- Menetapkan kontrol sesuai klasifikasi.
- Jangan menyimpan data sensitif yang tidak perlu.
- Mengenkripsi semua data sensitif pada *database*
- Menggunakan standar algoritma, protokol yang mutakhir dan kuat, serta menerapkan manajemen kunci yang tepat.
- Mengenkripsi semua data saat transmisi dengan protokol aman seperti TLS dengan *cipher perfect forward secrecy* (PFS), prioritas cipher oleh server, dan parameter yang aman. Menerapkan enkripsi seperti HTTP *Strict Transport Security* (HSTS).
- Menonaktifkan *caching* untuk respons yang berisi data sensitif.
- Menyimpan kata sandi (*password*) menggunakan fungsi *hashing* adaptif dan salted yang kuat.
- Verifikasi secara independen efektivitas konfigurasi dan pengaturan.
- Melakukan audit dan pemantauan keamanan.

A03:2021 - Injection

Deskripsi

Kerentanan muncul ketika data yang tidak terpercaya dikirimkan ke interpreter, yang dapat menyebabkan eksekusi perintah yang tidak diinginkan. Kerentanan disebabkan antara lain:

- Kurangnya validasi input
- Sanitasi data yang tidak cukup
- Penggunaan metode pengolahan data yang tidak aman
- Kueri secara dinamis atau permintaan yang tidak diberikan parameter tanpa konteks-peringatan pengalihan

Dampak

- Penyerang bisa menyisipkan kode berbahaya yang akan dijalankan oleh sistem secara ilegal.
- Kebocoran data sensitif.
- Kerusakan sistem dan data
- Kerugian Finansial
- Kerusakan reputasi
- Konsekuensi hukum dan kepatuhan

Mitigasi

- Penyimpanan data terpisah dari perintah dan *query*.
- Menggunakan API yang aman yang mencegah penggunaan mesin penerjemah secara keseluruhan, menyediakan sebuah tatap muka berparameter, atau migrasi ke perangkat pemetaan relasi objek.
- Menggunakan daftar (*whitelist*) pada validasi input di sisi server.
- Menerapkan validasi dan sanitasi terhadap input pengguna.
- Menggunakan LIMIT dan kontrol SQL lainnya di antara *query* untuk mencegah terungkapnya data jika terjadi injection.
- Melakukan pengujian keamanan

A04:2021 - Insecure Design Vulnerabilities

Deskripsi

Insecure design vulnerabilities adalah celah keamanan yang muncul akibat kurangnya perhatian atau perencanaan yang memadai pada aspek keamanan selama fase desain aplikasi atau sistem. Kerentanan ini terjadi ketika arsitektur dan perencanaan sistem tidak memperhitungkan ancaman potensial atau tidak mengikuti praktik keamanan yang terbaik.

Terdapat perbedaan antara desain tidak aman dan implementasi tidak aman. Sebuah desain aman masih bisa memiliki kerusakan implementasi yang mengarah ke kerentanan yang dapat dieksploitasi. **Suatu desain tidak aman tidak dapat diperbaiki oleh sebuah implementasi yang sempurna**. Satu dari faktor yang berkontribusi terhadap desain tidak aman adalah ketiadaan pembuatan profil risiko bisnis yang inheren dalam perangkat lunak atau sistem yang sedang dikembangkan, maka menjadi kegagalan untuk menentukan desain keamanan level yang diperlukan.

Dampak

Sistem menjadi rentan terhadap eksploitasi bahkan sebelum proses implementasi dan pengujian dimulai.

Mitigasi

- Membuat dan menggunakan prosedur pengembangan secara aman untuk mengevaluasi dan mendesain kontrol keamanan.
- Menggunakan permodelan ancaman untuk autentikasi darurat, kontrol akses, *business logic*, dan *key flows*.
- Mengintegrasikan kendali dan bahasa keamanan ke dalam *use case*.
- Mengintegrasikan pengujian untuk *frontend* dan *backend*.
- Mensegregasikan lapisan tier pada sistem dan lapisan jaringan berdasarkan kebutuhan eksposur dan proteksi
- Mensegregasikan tenant secara robust dengan desain pada seluruh tier
- Membatasi konsumsi sumber daya oleh pengguna atau layanan

A05:2021 - Security

Misconfiguration

Deskripsi

Security misconfiguration merupakan kondisi sistem komputer, aplikasi, atau infrastruktur IT tidak dikonfigurasi dengan baik untuk melindungi informasi sensitif dan sumber daya organisasi dari ancaman keamanan. Konfigurasi yang salah atau kurangnya pembaruan pada sistem dan aplikasi dapat menciptakan celah keamanan yang dapat dimanfaatkan oleh penyerang. Hal ini sering disebabkan oleh:

- Kelalaian dalam pengelolaan konfigurasi keamanan atau kurangnya pemahaman mengenai praktik keamanan yang terbaik.
- Tidak memiliki pertahanan yang sesuai atau *security hardening* yang diperlukan
- Fitur - fitur yang tidak digunakan masih di enable atau diinstall
- Menggunakan akun dan password default atau tidak pernah diubah.
- Cara menghandle error terlalu informatif kepada user
- Tidak menggunakan fitur keamanan terbaru.
- Pengaturan security pada server aplikasi, framework aplikasi tidak diatur secara aman.
- Server tidak mengirim *security header* atau directives, atau tidak diatur secara aman..
- Menggunakan software yang tidak *update* atau rentan.

Dampak

Eksploitasi terhadap celah konfigurasi.

Mitigasi

- Melakukan otomatisasi terhadap proses *hardening* untuk meminimalisir usaha yang diperlukan untuk mengatur *environment* baru yang aman.
- Menghapus atau tidak menginstall fitur dan framework yang tidak digunakan.
- Meninjau dan memperbarui konfigurasi yang sesuai dengan standar keamanan.
- Menerapkan segmentasi antar komponen dengan segmentasi, containerization, atau cloud security groups (ACLs).
- Menerapkan *security headers*.
- Melakukan automasi untuk memverifikasi keefektifan dari konfigurasi dan setting di semua environments.

A06:2021 - Vulnerable and Outdated Components

Deskripsi

Vulnerable and Outdated Components merupakan kerentanan perangkat lunak dalam aplikasi web yang memiliki celah keamanan atau tidak lagi mendapatkan dukungan berupa pembaruan atau *patch* dari pengembangnya. Komponen dapat mencakup *library*, *framework*, modul, atau bagian lain dari perangkat lunak yang digunakan dalam pengembangan aplikasi web.

Komponen yang tidak diperbarui dapat menjadi sasaran potensial bagi penyerang yang mencari celah keamanan untuk dieksploitasi. Kejadian ini terjadi antara lain karena:

- Komponen tersebut mungkin mempunyai bug atau celah keamanan yang telah diketahui dan dipublikasikan, tetapi belum diperbaiki.
- Tidak teridentifikasinya komponen yang digunakan secara langsung maupun dependensinya.
- Perangkat lunak rentan, tidak didukung, atau sudah usang.
- Tidak memindai kerentanan secara teratur terkait dengan komponen yang digunakan.
- Tidak memperbaiki atau mengupdate platform, kerangka kerja, dan dependensi yang digunakan.
- Tidak menguji kompatibilitas pustaka-pustaka yang diperbarui, ditingkatkan, atau di-patch.
- Tidak mengkonfigurasi komponen secara aman.

Dampak

Terdapat kerentanan/bug pada komponen yang dapat dieksploitasi

Mitigasi

- Menghapus dependensi, fitur, komponen, file, dan dokumentasi yang tidak digunakan.
- Inventarisasi versi komponen sisi klien dan sisi server secara terus menerus dan dependensinya
- Memantau secara terus menerus sumber-sumber informasi terkait kerentanan komponen seperti Common Vulnerability and Exposures (CVE) dan National Vulnerability Database (NVD) untuk kerentanan dalam komponen.
- Hanya mengunduh dan menginstal komponen dari sumber resmi melalui tautan yang aman. P

- Memantau pustaka dan komponen yang tidak dirawat atau tidak membuat patch keamanan untuk versi yang lebih lama.

A07:2021 - Identification and Authentication Failures

Deskripsi

Identification and authentication failures merupakan kelemahan atau kegagalan dalam sistem identifikasi dan autentikasi yang dapat menyebabkan akses tidak sah atau kebocoran informasi sensitif. Hal ini dapat terjadi ketika aplikasi tidak berhasil memverifikasi identitas pengguna atau tidak mengelola akses dengan efektif. Kerentanan ini disebabkan antara lain karena:

- Mengijinkan bruteforce / serangan otomatis seperti isian kredensial, di mana penyerang memiliki daftar nama pengguna dan kata sandi yang valid.
- Mengijinkan penggunaan kata sandi bawaan atau lemah.
- Menggunakan pemulihan (*restore*) kredensial yang lemah atau tidak efektif dan proses lupa kata sandi yang tidak aman.
- Menggunakan password dengan teks biasa, terenkripsi, atau dengan hash yang lemah
- Memiliki otentikasi multi-faktor yang hilang atau tidak efektif.
- Mengekspos ID Sesi di URL
- Tidak membatalkan ID Sesi dengan benar. Sesi pengguna atau token autentikasi (terutama token single sign-on (SSO)) tidak divalidasi dengan benar selama logout atau periode tidak aktif.

Dampak

Kegagalan dalam aspek ini dapat menciptakan celah bagi penyerang untuk mendapatkan akses yang tidak sah, menyusup ke dalam sistem, atau mengeksploitasi data pribadi.

Mitigasi

- Menerapkan otentikasi multi-faktor untuk mencegah pengisian kredensial otomatis, brute force, dan serangan penggunaan kembali kredensial yang hilang / dicuri.
- Tidak menggunakan kredensial bawaan apa pun, terutama untuk pengguna admin.
- Menerapkan pemeriksaan kata sandi (password) yang lemah, seperti menguji kata sandi baru.
- Menerapkan kata sandi (password) dengan mengatur panjang sandi, kompleksitas, dan kebijakan rotasi sesuai pedoman yang berlaku (misal NIST 800-63b)
- Pastikan pendaftaran, pemulihan (*restore*) kredensial, dan jalur API diperkuat terhadap serangan enumerasi akun dengan menggunakan pesan yang sama untuk semua hasil.
- Mencatat dan membatasi upaya login yang gagal.

- Menggunakan pengelola sesi built-in sisi server yang aman, menghasilkan ID sesi acak baru dengan entropi tinggi setelah login.
- ID sesi tidak boleh ada di URL, disimpan dengan aman, dan tidak valid setelah keluar, *idle*, dan waktu tunggu absolut.

A08:2021 - Software and Data Integrity Failures

Deskripsi

Software and Data Integrity Failures merupakan kondisi keaslian, konsistensi, dan keamanan perangkat lunak serta data tidak dapat dipastikan. Kegagalan integritas ini dapat disebabkan oleh berbagai faktor diantaranya:

- Kerentanan dalam source code, serangan siber, dan penggunaan komponen perangkat lunak yang tidak aman
- Kode dan infrastruktur yang tidak mencegah terjadinya pelanggaran integritas
- Aplikasi yang bergantung pada *plugins*, *libraries*, atau *modules* dari sumber yang tidak dipercaya dan *pipeline* yang tidak aman.
- Pembaruan otomatis yang diunduh tanpa adanya verifikasi integritas.

Dampak

Akses ilegal/tidak sah, kode yang berbahaya, atau kerusakan sistem.

Mitigasi

- Menggunakan tanda tangan digital atau mekanisme yang sama untuk memverifikasi bahwa perangkat lunak atau data berasal dari sumber yang diharapkan dan tidak dimanipulasi.
- Memastikan *libraries* dan dependensi seperti npm atau maven menggunakan repositori yang terpercaya.
- Memastikan keamanan rantai pasokan perangkat lunak dengan memverifikasi bahwa komponen tersebut tidak memiliki kerentanan yang sudah diketahui.
- Memastikan adanya proses review ketika mengubah kode dan konfigurasi untuk meminimalisir kemungkinan kode atau konfigurasi berbahaya masuk ke dalam *pipeline* perangkat lunak anda.
- Memastikan *CI/CD pipeline* anda memiliki metode pemisahan, konfigurasi dan akses kontrol yang tepat untuk memastikan integritas kode yang masuk mulai dari proses *build* / pembangunan hingga proses *deployment* / penyebaran perangkat lunak.
- Memastikan data yang belum di tandatangani atau tidak terenkripsi ini tidak terkirim ke klien yang tidak dipercaya tanpa adanya pengecekan integritas atau tanda tangan digital untuk mendeteksi apakah data telah di manipulasi atau pemutaran ulang data yang telah di serialisasi.

A09:2021 - Security Logging and Monitoring

- **Deskripsi**

Security Logging and Monitoring Failures terjadi ketika sistem tidak mampu mencatat (logging) dan memantau (monitoring) aktivitas keamanan dengan efektif

Dampak

- Kegagalan login dan transaksi dengan nilai yang tinggi tidak di catat.
- Peringatan dan Error tidak menghasilkan pencatatan yang memadai atau catatan pesan yang tidak jelas.
- Log dari aplikasi dan API tidak di monitor untuk aktifitas mencurigakan.
- Log hanya disimpan secara lokal.
- Threshold peringatan yang sesuai dan proses dari respon eskalasi tidak efektif
- Sistem tidak dapat melacak, menganalisis, dan merespons ancaman keamanan dengan cepat dan akurat.

Mitigasi

- Memastikan semua kesalahan login, kontrol akses dan validasi input dari server-side dapat di catat dan disimpan dengan waktu yang cukup untuk analisis forensik.
- Memastikan semua catatan dihasilkan dalam format dimana solusi pengelola catatan dapat dengan mudah digunakan.
- Memastikan data catatan telah di encode dengan benar untuk mencegah injeksi atau serangan pada pencatatan atau sistem monitor.
- Memastikan transaksi dengan nilai yang tinggi memiliki jejak audit dengan kontrol integritas untuk mencegah gangguan dan penghapusan, seperti hanya dapat ditambahkan ke database atau yang mirip seperti itu.
- Melakukan monitoring secara efektif dan memberikan peringatan terhadap aktifitas mencurigakan yang terdeteksi dan merespon secara cepat.
- Membuat atau adopsi sebuah respon insiden dan rencana pemulihan

A10:2021 - Server-Side Request Forgery

Deskripsi

Server-Side Request Forgery (SSRF) adalah jenis kerentanan keamanan yang terjadi ketika aplikasi web mengambil remote resource tanpa memvalidasi URL yang disediakan pengguna. Hal ini memungkinkan penyerang untuk memaksa aplikasi mengirimkan permintaan yang telah dimanipulasi ke tujuan yang tidak diharapkan, bahkan ketika aplikasi telah dilindungi oleh firewall, VPN, atau access control list (ACL) lainnya.

Dampak

- Penyerang dapat memaksa aplikasi untuk mengirim *crafted request* ke destinasi yang tidak diharapkan.
- Kebocoran data sensitif.
- Akses ke jaringan internal
- Serangan lateral dan eskalasi
- Gangguan / kerusakan sistem
- Pengungkapan infrastruktur dan arsitektur
- Pelanggaran kepatuhan

Mitigasi

- Melakukan segmentasi fungsionalitas akses ke remote resource dalam jaringan terpisah untuk mengurangi dampak SSRF.
- Menerapkan kebijakan firewall “deny by default” atau aturan kontrol akses jaringan yang hanya mengizinkan lalu lintas intranet penting.
- Sanitasi dan validasi semua data input pengguna.
- Menerapkan skema URL, port, dan tujuan dengan positive allow list.
- Menonaktifkan *HTTP redirections*.
- Perhatikan konsistensi URL untuk menghindari serangan seperti DNS rebinding dan kondisi race TOCTOU (time of check, time of use).
- Tidak mengembangkan layanan yang berhubungan dengan keamanan pada sistem yang berada di barisan depan,
- Khusus untuk *frontends* dengan pengguna/grup pengguna yang loyal atau berdedikasi serta dapat dikelola gunakanlah enkripsi jaringan (VPN) pada sistem independen mengingat adanya kebutuhan proteksi yang sangat tinggi.